Session 07

Application Confinement

Security of Information Systems (SIS)

Computer Science and Engineering Department

November 15, 2023

Papers

- Efficient software-based fault isolation
- Boxify: Full-fledged App Sandboxing for Stock Android

Story So Far

- systems and system components have an attack surface
- flaws in systems and system components may be exploited
- input may be used maliciously
- prevent existance and prevent exploitation of vulnerabilities
- defender needs to limit damage

Limiting Damage

- isolate entire system, e.g. virtualization
- isolate/confine system component (application), e.g. sandboxing
- limit possible actions, limit accessible resources, e.g. prevent an app from using the network, prevent an app from reading data from other apps

Application Confinement

- What can an application do? What can an application access?
- access control: subject, object
- typically enforced at kernel level
- What if it were enforced by a library at application level?
- overhead
- filesystem: users, file permissions, access control lists
- configurable permissions: Android permissions, iOS Privacy Settings, Linux capabilities
- sandboxing: jailing (filesystem), application sandboxing (kernel-enforced rules)

Remember: Malware

- application deployed on user device/workstation
- may abuse resource use and access
- doesn't require a vulnerability in an app, only a defect in the configuration or system
- confining it reduces damage

Filesystem Access Control

- subject: process (UID)
- object: file (UID, GID)
- permissions or access control lists (attached to a file)

Android Permissions

- requests permissions at runtime
- permission approval
- enforcement at Android SDK level
- signed permissions

iOS Privacy Settings

- database mappping between app and resource/service
- `Preferences` app writes to database
- may be turned on/off

Linux Capabilities

- security tokens providing privileges
- attached to a given process
- allow different permissions for processes belonging to the same user
- may also be attached to an executable (similar to the setuid bit)

Linux Security Modules

- framework in Linux kernel
- hooks for user-level system call
- introduced in Linux kernel 2.6

MAC Implementations

- SELinux (2.6.0)
- AppArmor (2.6.36)
- Smack (2.6.25)
- TOMOYO (2.6.30)
- Yama (3.4)

SELinux

- inode based
- uses labels - user:role:type:mls
- policy based
- modes
  - disabled
  - permissive
  - enforcing
- other features
  - Role-Based Access Control (RBAC)
  - Multi-Level Security (MLS)
  - Multi-Category Security (MCS)

AppArmor

- path based
- filesystem agnostic
- profile based
- hybrid modes
  - per object mode
  - learning mode

SMACK

- inode based
- uses labels (most are kept in extended attribute – `xattrs`)
- policy based
- access
  - rwxa - same as DAC
  - t - transmutation
  - b - report in bringup mode
- custom labels: `_` $\hat{*}$? `@`

Assets to Protect

- file descriptors
- file system space
- other processes
- memory
- network
- everything else

Sandbox Implementations

- capabilities
- jail
- rule based (MAC)
- Java Virtual Machine
- HTML5 iframe sandbox
- .NET Code Access Security

Breaking Sandboxing

- faulty sandbox rules
- other faulty configuration
- kernel vulnerability

Linux Seccomp

- minimize the exposed kernel surface
- to be used by developers
- uses BPF (*Berkeley Packet Filtering*)
- requires support in kernel

Kernel Config

- CONFIG_HAVE_ARCH_SECCOMP_FILTER=y
- CONFIG_SECCOMP_FILTER=y
- CONFIG_SECCOMP=y

Default Allowed Syscalls

- read
- write
- exit
- sigreturn

Android Application Sandbox

- *The sandbox is simple, auditable, and based on decades-old UNIX-style user separation of processes and file permissions.*
- SELinux-based
- uses application UID to map sandbox to application

Sandbox Profiles

- set of rules
- sandbox operations, sandbox filters
- provided as binary blobs in the kernel image
- attached to an application
- some apps may use the same sandbox profile
- some system services use no sandbox profile
- entitlement-checks and sandbox extensions for differentiation between apps using same sandbox profile

container Sandbox Profile

- default sandbox profiles for **all** 3rd party apps
- biggest sandbox profile

SandScout

- https://dl.acm.org/citation.cfm?id=2978336
- SandScout: Automatic Detection of Flaws in iOS Sandbox Profiles
- systematic analysis of container sandbox profiles
- found flaws: application collusion, device abuse, control bypass

Keywords

- access control
- Linux Security Module
- subject, object, permission
- capabilities
- profiles
- MAC
- SELinux, AppArmor, SMACK
- seccomp
- iOS sandboxing
- privacy settings